Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Withdrawn) A system, comprising:

a processor;

storage coupled to the processor and containing elements of metadata belonging to a plurality of schemas; and

mappings between the elements of metadata, each mapping being expressed as metadata and comprising a processor executable functional expression that relates the elements of metadata together.

2. (Withdrawn) The system of claim 1 wherein the elements of metadata comprise processor readable objects selected from the group consisting of resources, properties, and literals.

3. (Withdrawn) The system of claim 1 wherein the metadata comprises processor readable objects selected from the group consisting of dictionaries, catalogs, and directories.

4. (Withdrawn) The system of claim 1 wherein the functional expressions comprise processor readable parameters that represent a resource that aggregates a name, type, and parameter path.

5. (Withdrawn) The system of claim 1 wherein the functional expressions comprise processor readable parameters that represent a resource aggregating a type and a parameter path and that is connected to a name through an explicit mapping.

6. (Withdrawn) The system of claim 1 wherein a value of a previously calculated functional expression is cached in the storage.

7. (Withdrawn) The system of claim 1 wherein reasoning tasks are defined over the mappings.

8. (Withdrawn) The system of claim 1 further comprising processor readable dependency chains that define dependent relationships between properties of parameter paths of the functional expressions.

9. (Withdrawn) The system of claim 8 wherein the dependency chains are constructed using sub-properties of a transitive property that distinguishes dependency chains with common parameter subpaths.

10. (Withdrawn) The system of claim 8 wherein the dependency chains comprise dependency chains that are validated between the plurality of schemas.

11. (Currently amended) A method performed by at least one processor, the method comprising:

generating a node to represent a functional relationship between one or more objects of distinct ontologies in a metadata system;

associating a**[[**n**]]** metadata expression of the functional relationship to the node; and

associating one or more parameters of the functional relationship to the node.

12. (Original) The method of claim 11 further comprising associating a dependency chain representing the dependent relationships between properties of a parameter path associated with the one or more parameters of the functional relationship.

13. (Original) The method of claim 11 wherein associating one or more parameters comprises generating a resource that aggregates a local name, type, and dependency chain.

14. (Original) The method of claim 11 wherein associating one or more parameters comprises generating a resource that aggregates a type and a dependency chain and that is associated to a name through an explicit mapping.

15. (Original) The method of claim 11 further comprising identifying mappings between dependency chains spanning the distinct ontologies.

16. (Previously presented) The method from claim 15 wherein the identifying further comprises utilizing heuristics to suggest alternative mappings between dependency chains.

17. (Original) The method of claim 15 further comprising maintaining the mappings that span the distinct ontologies when one of the distinct ontologies is modified.

18. (Currently amended) A computer readable medium storing a program executable by a processor, the program causes the processor to:

generate a node to represent a functional relationship between one or more objects of distinct ontologies in a metadata system;

link to the node a**[[**n**]]** metadata expression of the functional relationship; and

link one or more parameters of the functional relationship to the node.

19. (Original) The computer readable medium of claim 18 wherein the program further causes the processor to connect a dependency chain representing the dependent relationships between properties of a parameter path.

20. (Original) The computer readable medium of claim 18 wherein the program further causes the processor to connect one or more parameters comprising generating a blank node that aggregates a local name, type, and dependency chain.

21. (Withdrawn) A system, comprising:

a means for executing instructions;

a means for storing elements of metadata belonging to a plurality of schemas; and

a means for mapping the elements of metadata, the means for mapping comprising processor readable functional expressions executable by the means for executing instructions.

22. (Withdrawn) The system of claim 21 wherein the elements of metadata comprise processor readable objects selected from the group consisting of resources, properties, and literals.

23. (Withdrawn) The system of claim 21 wherein the functional expressions comprise processor readable parameters representing the elements of metadata, the parameters comprising blank nodes that aggregate a name, type, and parameter path.

24. (Withdrawn) The system of claim 21 wherein the processor readable functional expressions comprise parameters representing the elements of metadata, the parameters comprising resources that are connected to a name through an explicit mapping.

25. (Withdrawn) The system of claim 21 wherein a value of a previously calculated functional expression is cached in the means for storing elements of metadata.